

Method and Apparatus for Image Processing
Capable of High Speed Processing

[0001] This application is based on application No. 2001-1370 filed in Japan, the content of which is hereby incorporated by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention relates to method and apparatus for image processing and, more particularly, to the claimed method and apparatus for image processing capable of high speed processing.

Description of the Related Art

[0003] An image processing apparatus or a program executing an error diffusion process has been known.

[0004] Fig. 15 is a block diagram representing a configuration of a conventional image processing apparatus employing the error diffusion method.

[0005] Referring to the figure, the image processing apparatus includes an input unit 501 successively receiving as inputs image signals (input values) representing density of respective pixels; a subtractor 503 reading an error value corresponding to a pixel of interest from an error memory 513 and subtracting the error value from the input value; a thresholding unit 507 thresholding, with a prescribed threshold value, an output (modified input) of subtractor 503; an output unit 509 outputting the result of thresholding (for example, the value 0/1); a subtractor 511 subtracting the output of subtractor 503 from the output of thresholding unit 507; and the error memory 513 distributing the output of subtractor 511 as an error to surrounding pixels and stores.

[0006] By the image processing apparatus configured as described above, an error in pixel density resulting from gradation reducing process is diffused. Therefore, it becomes possible to reduce gradations of an image as an object of processing, while maintaining the density of the image as a whole.

[0007] The conventional image processing apparatus, however, has a

problem that the speed of processing is low.

SUMMARY OF THE INVENTION

[0008] Therefore, an object of the present invention is to provide a method and an apparatus for image processing capable of high speed image processing.

[0009] In order to attain the above described object, an aspect of the present invention provides an image processing apparatus, including: an input unit successively receiving, as inputs, image signals representing pixels; a determining unit determining whether an input signal represents a white pixel; and an error diffusion processing unit performing different processes depended on whether the input signal represents a white pixel or not.

[0010] Preferably, the error diffusion processing unit outputs a signal representing a white pixel, and does not perform subsequent calculation of error and distribution of the error to pixels, when the input signal represents a white pixel.

[0011] Preferably, the error diffusion processing unit performs the error diffusion process using a threshold value that is smaller than a central value of values that can be assumed by the image signals.

[0012] Preferably, the error diffusion processing unit changes the threshold value in relation to the magnitude of the signal input through the input unit.

[0013] Preferably, the error diffusion processing unit performs a process for subtracting a prescribed value before distributing the calculated error, and adding a prescribed value before thresholding.

[0014] According to another aspect, the present invention provides an image processing apparatus, including: an input unit successively receiving, as inputs, image signals representing pixels; a determining unit determining whether an input signal represents a black pixel; and an error diffusion processing unit performing different processes depended on whether the input signal represents a black signal or not.

[0015] Preferably, the error diffusion processing unit outputs a signal representing a black pixel, and does not perform subsequent calculation of

error and distribution of the error to pixels, when the input signal represents a black pixel.

[0016] According to a still further aspect, the present invention provides a method of image processing, including the steps of: successively
5 inputting image signals representing pixels; determining whether an input signal represents a white pixel or a black pixel; and error diffusion process step performing different processes dependent on whether the input signal represents either white or black pixel or not.

[0017] According to another aspect, the present invention provides a
10 recording medium recording an image processing program, for realizing the above described method.

[0018] According to another aspect, the present invention provides a method of image processing, including the steps of: successively inputting
15 image signals representing pixels; determining whether an input signal represents a white pixel; and error diffusion process step performing different processes dependent on whether the input signal represents a white pixel or not.

[0019] The foregoing and other objects, features, aspects and
20 advantages of the present invention will become more apparent from the following detailed description of the present invention when taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Fig. 1 is a block diagram representing a hard ware configuration
25 of an image processing apparatus in accordance with a first embodiment of the present invention.

[0021] Fig. 2 is a flow chart representing the process in accordance with the first embodiment.

[0022] Fig. 3 is an illustration representing an operation of the error
memory.

[0023] Fig. 4 is an illustration representing the effect of the first
30 embodiment.

[0024] Fig. 5 shows the result of image processing in accordance with the first embodiment.

[0042] Fig. 23 shows the result of image processing with the error memory cleared when the input is "0".

[0043] Fig. 24 shows the result of image processing with the value of the error memory maintained when the input is "0".

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0044] First Embodiment

[0045] Most of the images (particularly, images of business documents) are largely occupied by white pixels. When half tone processing is performed using the error diffusion method, the speed of image processing would be significantly improved if distribution of error is stopped for white pixels.

[0046] Therefore, in the image processing apparatus in accordance with an embodiment of the present invention, a normal error diffusion process is performed when the input is not "0" (white), and error diffusion is not performed but "0" is output when the input is "0". By adopting such a configuration, the speed of image processing can significantly be improved. By the following approach, image quality can also be improved.

[0047] In such a structure as described above, in which normal error diffusion process is performed when the input is not "0" and error diffusion is not performed but "0" is output when the input is "0", there arises a problem of how to address the errors accumulated in the error memory.

Assume, for example, that an image including white pixels (pixels of which input value is "0") is processed by the conventional error diffusion method.

[0048] When the input is close to "0", an average value of error values becomes approximately equal to the threshold value. When the error value of the pixel of interest in the error memory is cleared at this time, there would be a contradiction in the relation with other pixels. When the error value of the pixel of interest in the error memory is left as it is, the error will be distributed to an unrelated pixel (for example, a pixel at a far distance).

[0049] How the aforementioned program occurs when the input is close to "0" will be described in detail, with reference to the figures.

[0050] Referring to Fig. 16, assume that the input may be in the range of 0 to 1. The threshold value is represented by th . Further, it is assumed

that pixel values are input successively to the image processing apparatus starting from pixel (1), and subjected to binarization process. When the input is ϵ (a small value not "0"), "0" is output in the processing of pixels (1) to (4). During this process, errors are accumulated, and therefore, a
5 modification input (output of subtractor 503 shown in Fig. 14) increases gradually.

[0051] As the modification input exceeds the threshold value th at pixel (5), "1" is output, and the error decreases. By repeating this operation, the error in the error memory comes to have a value close to the threshold value
10 th and stabilized.

[0052] When the input changes to "0" as represented by (14) in Fig. 17 after the stable state is attained, it will be input = output = "0", and therefore the accumulated errors are maintained as they are until the input changes (until pixel (18)).

15 [0053] This is true no matter how many pixels of which input is "0" continues. When the input assumes a value other than "0" (after pixel (18)), the modification input still remains close to the threshold value th . Therefore, a dot is output immediately (pixel (19)).

20 [0054] When distribution of the error is stopped and the error memory is cleared when the input assumes "0" as represented by (14) and the following of Fig. 18, the error will be "0", and hence input = modification input = output = "0".

[0055] When the input again returns to ϵ as represented by the pixel (18), accumulation of errors start again from this state. No dot is output
25 until the modification input exceeds the threshold value th . Therefore, generation of a dot delays.

[0056] Similarly, referring to Fig. 19, when the input value is $1 - \epsilon$, which is large but not 1, the error in the error memory comes to have a value close to the threshold value th and stabilized.

30 [0057] Referring to Fig. 20, if the error memory is cleared when the input assumes "1", the error will be "0". Thereafter, when the input again assumes $1 - \epsilon$, the state of "no dot" is not attained until the modification input exceeds the threshold value (becomes equal to or lower than the

threshold value). Therefore, generation of a white pixel delays.

[0058] As described above, when the error memory is cleared, generation of a dot (or generation of a white pixel) delays.

[0059] It seems that such a problem can be solved simply by not clearing the error memory. When the error memory is not cleared, however, errors generated from the pixels would be distributed to unrelated pixels, resulting in an image of lower quality.

[0060] Fig. 21 shows a specific image sample as the object of processing. This image includes a plurality of lines extending in a direction from upper left to lower right. Further, background of the sample image has a gradation, which is denser at a lower portion.

[0061] Fig. 22 shows the result of processing of the sample image shown in Fig. 21 by the conventional error diffusion process. In this example, the original image can be reproduced fairly satisfactory. The speed of image processing, however, is low, as error diffusion process is performed on white pixels in the sample image also.

[0062] Fig. 23 shows the result of processing of the sample image shown in Fig. 21, in which error diffusion is not performed when the input (pixel value) is "0" and the error memory is cleared.

[0063] As already described with reference to Fig. 18, once the input assumes "0", dot is not output for a while thereafter, and therefore, reproductivity of the original image is not satisfactory in this example.

[0064] Fig. 24 shows the result of image processing where error diffusion is not performed when the input (pixel value) is "0" and the value of the error memory is kept as it is. In this example, errors of pixels not related at all are distributed in the error diffusion process, and therefore, the original image cannot be well reproduced either.

[0065] The present embodiment provides an image processing apparatus that solves the above described problem and enables high speed image processing without degrading image quality. For this purpose, in the image processing apparatus of the present embodiment, whether the input is "0" (white pixel) or not is determined, and different processes are performed dependent on whether it is "0" or not.

[0066] More preferably, when the input is not "0", the normal error diffusion process is performed, and when the input is "0", the value "0" is output directly and the error memory is cleared, so that calculation of error and subsequent distribution of the error to pixels are not performed.

[0067] Further, a threshold value that is smaller than the central value of possible values of the input signal (threshold value used in the normal error diffusion process) is used, in order to prevent disadvantages resulting from not distributing the error.

[0068] Fig. 1 is a block diagram representing a hardware configuration of the image processing apparatus in accordance with the first embodiment of the present invention.

[0069] Referring to the figure, the image processing apparatus includes: a CPU 301 controlling the overall apparatus; a printer 303 for outputting an image; a display 305 for displaying an image; an LAN or modem card 307 for communication with external apparatuses; an input apparatus 309 such as a keyboard or a mouse, a floppy disk drive 311; a CD-ROM drive 313; a hard disk drive 315; an RAM 317; and an ROM 319.

[0070] A program or the image processing apparatus to execute the method of image processing may be provided by a floppy disk F inserted to floppy disk drive 311, a CD-ROM 313a inserted to CD-ROM drive 313 or the like. Such an image processing program may also be obtained through LAN or modem card 307.

[0071] Fig. 2 is a flow chart representing the process performed by the image processing apparatus shown in Fig. 1.

[0072] Referring to the figure, the image processing apparatus determines whether the input (pixel value) is "0" or not. If the input is not "0", image processing using the normal error diffusion method is performed in step S101.

[0073] More specifically, in step S101, the processes are performed in which an error of the pixel of interest is obtained from the error memory, and the error memory of the pixel of interest is reset accordingly. Further, the value of the input plus the obtained error (modification input) is thresholded, and 0 or 1 is output. Thereafter, the modification input is

subtracted from the output, and the result is distributed as an error to error memories of surrounding pixels. Thereafter, processing of the next pixel starts.

[0074] When the input is "0", the output is set to "0" (white pixel) in step S103. Calculation and distribution of the error are not performed, and the error memory of the pixel of interest is reset.

[0075] Fig. 3 is an illustration representing the operation of the error memory (in the present embodiment, corresponding to RAM 317 of Fig. 1).

[0076] Referring to the figure, when we represent the position in the error memory corresponding to the pixel as an object of processing is represented by * a, in the process shown in step S101 of Fig. 2, first, the error recorded at the position * a of the error memory is read. Then, the content of the position * a of the error memory is cleared (reset).

[0077] After the input + read error (= modification threshold value) is thresholded by the threshold value th and a value obtained by subtracting the modification threshold value from the result of thresholding is regarded as the error resulting from the processing of the pixel of interest, it is recorded (added to the already recorded value) at positions (for example, b to m) of the surrounding pixels in the error memory, in accordance with a prescribed weight coefficient.

[0078] When the input is "0" (step S103 of Fig. 2), the output is set to "0". The position (* a) corresponding to the pixel which is the object of processing in the error memory is cleared, and error propagation is not performed.

[0079] Referring to Fig. 4, in the present embodiment, the threshold value th is set smaller than in the prior art. Therefore, when the input is small, the error is stabilized at a small value. Therefore, the influence is limited even when the error memory is cleared.

[0080] When the input changes from a small value ε to 0 (pixel (4) of Fig. 4), the error memory is reset (pixel (5)). When the input again assumes a value other than 0 (pixel (8)), however, the modification input immediately exceeds the threshold value th. Thus, the delay in dot generation can be eliminated.

[0081] Even when the threshold value th is set small, density can

correctly be reproduced in the error diffusion method.

[0082] The threshold value may be set small, only when the input is small (equal to or smaller than a prescribed value).

[0083] Fig. 5 shows the result of processing the image of Fig. 21 in accordance with the present invention, where the threshold value th is set to $th = 0.2$.

[0084] As can be seen from the figure, according to the present embodiment, the problem (Fig. 23) experienced when the error memory is cleared can be solved, as the threshold value is set small.

[0085] Further, according to the present embodiment, distortion (especially, see an upper edge portion of the image) at the peripherally of the image can be reduced as compared with the processing (Fig. 22) in accordance with the conventional error diffusion method.

[0086] Second Embodiment

[0087] Fig. 6 is a block diagram representing a configuration of the image processing apparatus in accordance with the second embodiment of the present invention.

[0088] Referring to the figure, the image processing apparatus includes, in addition to the configuration of the image processing apparatus shown in Fig. 15, an adder 603 adding a constant k to the value read from error memory 513, a subtractor 605 subtracting the constant k from the error prior to the storage in error memory 513, and an output unit 601 outputting the constant k .

[0089] In the present embodiment also, as in the processing shown in Fig. 2, the normal error diffusion process is performed (S101) when the input is not "0", an 0 is output and the error memory of the pixel of interest is reset (S103) when the input is "0".

[0090] When the input is small, in order that the value in the error memory 513 is stabilized at a small value, a constant k is subtracted before the error is stored in the error memory, in the present embodiment. Therefore, the value stored in the error memory can be made smaller.

[0091] When the error is taken out from the error memory and used, the constant k is added. Therefore, correct error diffusion is possible.

[0092] Particularly, when the input is small, the value in the error memory when the input is small can be reduced (to almost 0) as shown in Fig. 7, by setting the absolute value of k to be slightly smaller than th . Therefore, even if the error is not distributed and the memory is cleared when the input is 0, the influence is limited.

[0093] Fig. 8 shows the result of processing the image of Fig. 21, when the constant $k = -0.45$ and threshold value $th = 0.5$, in the present embodiment. In the present embodiment also, the value in the error memory can be made small, and therefore, problems experienced when the memory value is cleared can be minimized.

[0094] Further, as compared with the processing (Fig. 22) of the conventional error diffusion method, distortion at the peripheral portion of the image can be suppressed also by the present embodiment.

[0095] Third Embodiment

[0096] The present invention is also applicable to the image processing apparatus using the threshold value diffusing process proposed by the applicant of the present application in Japanese Patent Laying-Open No. 2000-165669 (U.S.S.N. 09/399,180).

[0097] Fig. 9 is a block diagram representing a configuration of the image processing apparatus using the threshold value diffusing process.

[0098] Referring to the figure, the image processing apparatus includes: an input unit 121 receiving pixel values as inputs; an output unit 123 outputting the binarized value of 0 or 1; and a thresholding processing unit 101 performing thresholding operation on the input value input through input unit 121, based on the threshold value.

[0099] The image processing apparatus further includes: an output unit 107 outputting an initial threshold value th ; a subtractor 103 subtracting an error of a weighted and distributed threshold value from the initial thresholded value th ; an inverting unit 105 inverting the result of thresholding; a subtractor 109 subtracting an output of subtractor 103 from the output of inverting unit 105; a β multiplication unit 111 multiplying the output of subtractor 109 by a coefficient β ; and a weight dispersing memory 113 for dispersing the output of β multiplication unit 111 to the threshold values

corresponding to surrounding pixels.

[0100] In the error diffusion process, the difference between the modification input and the output is distributed as an error, to surrounding pixel values. In the threshold value diffusing process, the difference between an inverted value of the output and the threshold value used for processing the pixel is dispersed to the threshold values for processing surrounding pixels.

[0101] Fig. 10 is a block diagram representing the configuration of the image processing apparatus in accordance with the third embodiment of the present invention.

[0102] Referring to the figure, in the present embodiment, the image processing apparatus includes, in addition to the configuration of the apparatus employing the threshold value diffusing method shown in Fig. 9, a k output unit 201 outputting a constant k, an adder 205 adding the constant k to the output of β multiplication unit 111, and a subtractor 203 subtracting the constant k when a value is read from the weight dispersing memory 113.

[0103] In the present embodiment also, when the input is "0", "0" is automatically output, and the memory is cleared, and therefore the error in binarization is not distributed. In the present embodiment, however, a constant k is added when a value is recorded in the weight dispersing memory 113, and the constant k is subtracted when a value is output from the weight dispersing memory 113. Therefore, the value in the weight dispersing memory 113 can be made smaller, and therefore the influence caused by clearing the contents thereof can be reduced.

[0104] Fourth Embodiment

[0105] The hardware configuration of the image processing apparatus in accordance with the fourth embodiment is the same as that of the first embodiment. Therefore, description thereof will not be repeated.

[0106] In the fourth embodiment, in place of the process shown in the flow chart of Fig. 2, the process shown in the flow chart of Fig. 11 is executed.

[0107] More specifically, referring to Fig. 11, when the input is not "1",

embodiment

normal error diffusion process is performed. When the input is "1" (black pixel), "1" is automatically output, and the error memory of the pixel of interest is reset. Therefore, error diffusion process need not be performed when the input is "1". Thus, the speed of image processing is improved.

[0108] In the present embodiment, the threshold value th is set larger than the central value of the input values, as shown in Fig. 12. As the threshold value is set larger, even when the accumulated errors are reset when the input changes from a state close to "1" to "1", the influence thereof (delay in generating a white pixel) can be minimized. More specifically, when the input changes from "1" to a value not "1", the modification input immediately exceeds the threshold value th (becomes equal to or lower than the threshold value th), and therefore, the delay in generating a white pixel can be prevented.

[0109] Fifth Embodiment

[0110] The hardware configuration of the image processing apparatus in accordance with the fifth embodiment, is the same as that of the image processing apparatus in accordance with the first embodiment. Therefore, description thereof will not be repeated.

[0111] In the image processing apparatus in accordance with the fifth embodiment, in place of the process shown in Fig. 2, the process shown by the flow chart of Fig. 13 is executed.

[0112] More specifically, when the input is neither "0" nor "1", the normal error diffusion is performed. By contrast, when the input is "0", the process similar to the first embodiment is performed, in which "0" is output and the error memory of the pixel of interest is reset. When the input is "1", "1" is output and the error memory of the pixel of interest is reset, in the similar manner as the fourth embodiment.

[0113] More specifically, in the present embodiment, when the input is "0" or "1", the error diffusion process is not performed, and the error memory is reset.

[0114] Further, in the present embodiment, the threshold value is changed in accordance with the input, so that a proportional relation holds between the input and the threshold value, as shown in Fig. 14. Therefore,

when the input is small, the threshold value can be made smaller accordingly, and when the input is large, the threshold value can be made large accordingly. Therefore, when the input is small, the threshold value is also small, and therefore even when the error is not distributed but cleared when the input is "0", the influence is limited, as in the process shown in Fig. 4.

[0115] On the contrary, when the input is large, the threshold value is also large, and therefore even when the error is not distributed but cleared when the input is "1", the influence is limited as in the example of Fig. 12.

[0116] Effects of the Embodiment

[0117] As described above, in the above described embodiments, the image processing apparatus is configured such that when the input is close to "0" or "1", the distributed error value becomes small. Further, error diffusion process is not performed when the input is "0" or "1". Therefore, it is possible to significantly reduce the processing time for images which is largely occupied by the pixels of "0", such as general business documents. Further, the phenomenon of distorted contour of the image can simultaneously be solved.

[0118] The processing of the above described embodiments may be performed by a software or by using hardware circuitry.

[0119] The program for executing the process of the above described embodiments may be provided separately, and the program may be provided to the user recorded on a recording medium such as a CD-ROM, a floppy disk, a hard disk, ROM, RAM, a memory card or the like.

[0120] Further, the present invention is applicable both in a system connected to a network and a system not connected to a network environment.

[0121] Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.